PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-241777
(43)Date of publication of application : 21.09.1993

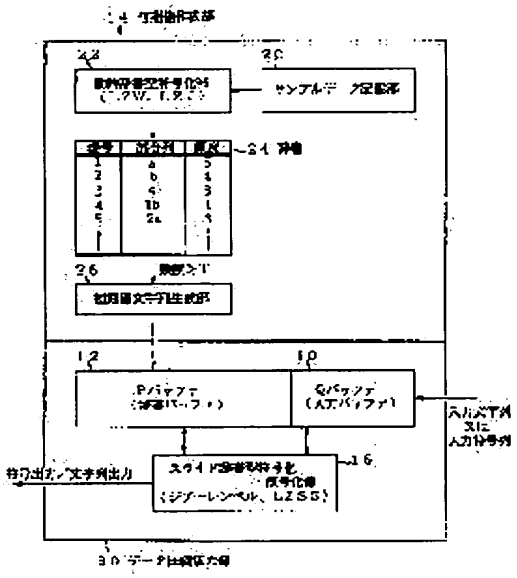
(51)Int.Cl. G06F 5/00
G06F 15/20
H03M 7/30

(21)Application number : 04-042578
(22)Date of filing : 28.02.1992

(71)Applicant : FUJITSU LTD
(72)Inventor : YOSHIDA SHIGERU
OKADA YOSHIYUKI
NAKANO YASUHIKO
CHIBA HIROTAKA

(54) DATA COMPRESSION SYSTEM

(57)Abstract:
PURPOSE: To improve the compressibility without spoiling the easiness of slide dictionary type algorithm by registering an initial value character string consisting of a special kind of data which is high in appearance frequency in a dictionary.
CONSTITUTION: An initial value generation part 14 encodes representative sample data according to the dynamic dictionary type algorithm. Then a counter counts the frequency of use of a reference number, indicating a character string registered in a dictionary generated by this encoding, at the time of the encoding as an appearance frequency, and registered character strings in the dictionary 24 which have appearance frequencies larger than a specific threshold value are extracted and arrayed when the encoding of the sample data ends to generate an initialization character string. The initialization value character string generated by the initialization value generation part 14 is used for data compression using a slide dictionary at a data compression and restoration part 30. Thus, the character strings which are high in use frequency are previously registered in the dictionary, so the probability that long input data match a character string to the longest length becomes high to improve the efficiency of the encoding.



LEGAL STATUS

[Date of request for examination] 13.03.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3241788
[Date of registration] 19.10.2001
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

P-2270

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平5－241777

(43)公開日　平成５年(1993)９月21日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　5/00		Ｈ　9189－5Ｂ		
15/20		５７０　Ｃ　7343－5Ｌ		
Ｈ０３Ｍ　7/30		8836－5Ｊ		

審査請求　未請求　請求項の数７(全 23 頁)

(21)出願番号　特願平4－42578

(22)出願日　平成４年(1992)２月28日

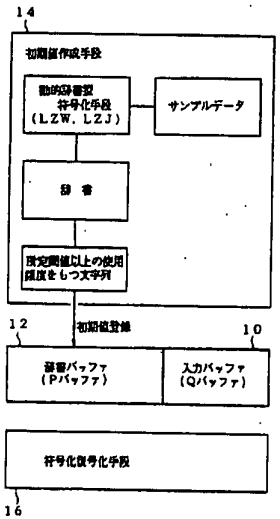
(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　吉田　茂
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　岡田　佳之
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　中野　泰彦
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　竹内　進　(外１名)

最終頁に続く

(54)【発明の名称】　データ圧縮方式

(57)【要約】
【目的】スライド辞書を用いたＬＺＳＳ符号等によるデータ圧縮方式に関し、スライド辞書型アルゴリズムの効率化を目的とする。
【構成】代表的なサンプルデータをＬＺＷ符号等の動的辞書型アルゴリズムで符号化して辞書に登録した文字列の使用頻度を求め、所定閾値以上の使用頻度をもつ登録文字列から初期値文字列を作成してスライド辞書に初期登録して符号化を開始する。

本発明の原理説明図



【特許請求の範囲】
【請求項１】入力バッファ（１０）中の入力データを辞書バッファ（１２）中の符号化済データの部分列の内、最長一致するものの格納位置と一致長で指定して符号化し、符号化済みの入力データは辞書バッファ（１２）に移して新たな符号化済みデータとして次の入力データを符号化するデータ圧縮方式に於いて、
代表的なサンプルデータを相異なる部分列に分けたとき、所定閾値以上の出現頻度をもつ部分列を抽出し、該抽出した部分列を一列に並べて初期値文字列を予め作成する初期値作成手段（１４）と、
該初期値作成手段（１４）で作成した初期値文字列を、符号化又は復号化に先立って最初に前記辞書バッファ（１２）に固定的に設定して符号化済又は復号化済データと見做し、該初期値文字列および新たに設定したび符号化済又は復号化済データの中から前記入力バッファ（１０）の入力データと最長一致する部分列を検索して格納位置と一致長で指定して符号化又は復号化する符号化復号化手段（１６）と、を備えたことを特徴とするデータ圧縮方式。
【請求項２】入力バッファ（１０）中の入力データを辞書バッファ（１２）中の符号化済データの部分列の内、最長一致するものの格納位置と一致長で指定して符号化し、符号化済みの入力データは辞書バッファ（１２）に移して新たな符号化済みデータとして次の入力データを符号化するデータ圧縮方式に於いて、
代表的なサンプルデータを相異なる部分列に分けたとき、所定閾値以上の出現頻度をもつ部分列を抽出し、該抽出した部分列を一列に並べて初期値文字列を予め作成する初期値作成手段（１４）と、
該初期値作成手段（１４）で作成した初期値文字列を、符号化に先立って最初に前記辞書バッファ（１２）に固定的に設定して符号化済データと見做し、該初期値文字列の中からのみ前記入力バッファ（１０）の入力データと最長一致する部分列を検索して格納位置と一致長で指定して符号化又は復号化する符号化復号化手段（１６）と、
を備えたことを特徴とするデータ圧縮方式。
【請求項３】請求項１，２記載のデータ圧縮方式に於いて、前記初期値作成手段（１４）は符号化済み文字列を参照番号を付して登録する辞書を有し、代表的なサンプルデータの文字列に最長一致する前記辞書中の符号化済み部分列を検索して参照番号で指定して符号化し、該符号化後に該参照番号に次のサンプル文字を付加した部分列を新たな参照番号を付して前記辞書に登録し、更に辞書に登録された符号化済み文字列の検索する毎に使用回数を計数し、前記サンプルデータの符号化終了した際に前記使用回数が所定閾値以上となる文字列を抽出し、該抽出した文字列を出現順に並べて初期値文字列を予め作成することを特徴とするデータ圧縮方式。
【請求項４】請求項１，２記載のデータ圧縮方式に於いて、前記初期値作成手段（１４）は符号化済み文字列を参照番号を付して登録する辞書を有し、代表的なサンプルデータの文字列に最長一致する前記辞書中の符号化済み部分列を検索して参照番号で指定して符号化し、該符号化後に符号化した入力文字列の各文字を順次接頭部分列とし、該接頭部分列に辞書中の部分列を加えた一定長の部分列を複数作成して全て辞書に登録し、更に辞書に登録された符号化済み部分列の検索する毎に使用回数を計数し、前記サンプルデータの符号化終了した際に前記使用回数が所定閾値以上となる部分列を抽出し、該抽出した部分列を出現順に並べて初期値文字列を予め作成することを特徴とするデータ圧縮方式。
【請求項５】請求項１，２記載のデータ圧縮方式に於いて、前記初期値作成手段（１４）は入力バッファ中の代表的なサンプルデータを辞書バッファ中の符号化済データの部分列の内、最長一致するものの格納位置と一致長で指定して符号化し、該符号化データを相異なる部分列に分けたとき、所定閾値以上の出現頻度をもつ部分列を抽出し、該抽出した部分列を出現順に並べて初期値文字列を予め作成することを特徴とするデータ圧縮方式。
【請求項６】請求項１，２記載のデータ圧縮方式に於いて、前記辞書バッファ（１２）を読出専用の第１メモリと、読出及び書込可能な第２メモリで構成し、前記初期値作成手段（１４）で作成した初期値文字列を前記第１メモリに固定行に記憶し、前記入力バッファ（１０）の符号化済み入力データは第２メモリに移して記憶することを特徴とするデータ圧縮方式。
【請求項７】請求項１，２記載のデータ圧縮方式に於いて、前記辞書バッファ（１２）を書替え禁止領域を設定した読出及び書込可能なメモリで構成し、符号化開始時に前記初期値作成手段（１４）で作成した初期値文字列を前記辞書バッファ（１２）の書替え禁止領域にロードした後に符号化を開始することを特徴とするデータ圧縮方式。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、スライド辞書を用いたジブ－レンペル符号化によるデータ圧縮方式に関し、特にスライド辞書を用いた符号化に動的辞書を用いた符号化を利用したデータ圧縮方式に関する。近年、文字コード、ベクトル情報，画像など様々な種類のデータがコンピュータで扱われるようになっており、扱われるデータ量も急速に増加してきている。
【０００２】大量のデータを扱うときは、データの中の冗長な部分を省いてデータ量を圧縮することで、記憶容量を減らしたり、速く伝送したりできるようになる。様々なデータを１つの方式でデータ圧縮できる方法としてユニバーサル符号化が提案されている。ここで、本発明の分野は、文字コードの圧縮に限らず、様々なデータに

適用できるが、以下では、情報理論で用いられている呼称を踏襲し、データの1ワード単位を文字と呼び、データが任意ワードつながったものを文字列と呼ぶことにする。

【0003】ユニバーサル符号の代表的な方法として、ジブーレンペル（Ziv-Lempel）符号がある（詳しくは、例えば、宗像「Ziv-Lempelのデータ圧縮法」，情報処理，Vol. 26，No. 1，1985年を参照のこと）。ジブーレンペル符号では

①スライド辞書型（ユニバーサル型ともいう）と、

②動的辞書型（増分分解型ともいう）

の2つのアルゴリズムが提案されている。これらの方式の実用向きの改良方法が発表され、補助記憶装置のファイル圧縮や、パソコン通信でのデータ伝送に利用されるようになっている。

【0004】

【従来の技術】まず従来のスライド辞書型のアルゴリズムと動的辞書型のアルゴリズムについて説明する。

（1）スライド辞書型アルゴリズム

このアルゴリズムは、演算量は多いが、高圧縮率が得られる方法である。即ち、符号化データを、過去のデータ系列の任意の位置から一致する最大長の系列に区切り（部分列）、過去の文字列の複製として符号化する方法である。

［ジブーレンペル符号］図10にスライド辞書型であるジブーレンペル符号の符号器の原理図を示す。

【0005】図10において、辞書バッファとしてのPバッファ12には符号化済みの入力データが格納されており、入力バッファとしてのQバッファ10にはこれから符号化するデータが入力されている。Qバッファ10の文字列をPバッファ12の文字列と照合し、Pバッファ12の中で一致する最大長の文字部分列を求める。そして、Pバッファ12中でこの最大長文字列を指定するため次の情報の組を符号化する。

【0006】

【表1】

Pバッファ中の最長一致系列の開始位置	一致する長さ	不一致のシンボル

【0007】次にQバッファ10内の符号化した文字列をPバッファ12に移して、新たなデータを得る。以下、同様の操作を繰り返し、データを部分列に分解して符号化する。すなわちジブーレンペル符号では、現在の文字コードの系列を、符号化済の過去の系列からの複製として符号化するものである。ジブーレンペル符号を用いた場合、文字コードの文書情報は1／2程度に圧縮できる。

【0008】更にスライド辞書型アルゴリズムの改良として、LZSS符号がある（T. C. Bell, "Better OPM/L Text Compression", IEEE Trans. on Commun., Vol. COM-34, No. 12, Dec. 1986参照）。LZSS符号では、［Pバッファ中の最大一致系列の開始位置］と［一致する長さ］の組と、［次のシンボル］とをフラグ区別して、符号量の少ない方で符号化する。

【0009】更にスライド辞書型アルゴリズムの改良として、1／4インチ・カートリッジ磁気テープの標準圧縮方式であるQIC-122符号がある。LZSS符号について次に説明する。

［LZSS符号］LZSS符号による符号化の処理フローを図11に示し、その原理図を図12及び図13に示すに示す。

【0010】LZSS符号による符号化は、図12（b）に示すように例えば4ビットのインデックス情報をもって、これから符号化する文字列を格納する例えば4ビットのインデックス情報に対応して16個の文字数を格納できるQバッファ10と、図12（a）に示すように、例えば12ビットのインデックス情報をもって4096個の符号化済の文字列を格納するPバッファ12とを備えるようにして構成する。

【0011】符号化処理は図11のフローチャートに示すように、ステップS1でPバッファ12を空にしてQバッファ10に入力データを詰めた後に、ステップS2でQバッファ10の文字列とPバッファ12の文字列とを照合し最長一致する文字部分列を求め、ステップS3で2文字以上であることを条件にステップS5に進んで、求められた文字部分列を指定するために

［文字列Sの出現位置］

［一致長］の組で符号化する。

【0012】続いてステップS6でQバッファ10内の符号化した文字列をPバッファ12に移して、Qバッファ10内に符号化した文字列分の新たな文字列を入力していくことで符号化を実行する。尚、最長一致文字部分列が1バイトのときは生データで符号化した方が有利であるので、ステップS4で

［生データ1バイト］をそのまま出力する。

【0013】更に、図13に示すように、8個の符号化データもしくは生データを1組のデータとしてまとめると共に、まとめられた各8個のデータが符号化データなのか生データなのかを示すステップS4，S5で得られたフラグビットでなる8ビット識別データを先頭に付加し、1組のデータとして出力する。

（2）動的分解型（増分分解）アルゴリズム

このアルゴリズムは、圧縮率はユニバーサル型より劣るが、シンプルで、計算も容易であることが知られている。

【0014】増分分解型ジブーレンペル符号では、入力シンボルの系列を

$X = a\,a\,b\,a\,b\,a\,b\,a\,a\,\cdots$

とすると、成分系列$X = X_0\ X_1\ X_2\ \cdots$への増分分解は次のようにする。まず$X_1$を既成分の右端のシンボルを取り除いた最長の列とし、

$X = a \cdot ab \cdot aba \cdot b \cdot aa \cdot \cdots$

となる。従って、

$X_0 = \lambda$（空列）

$X_1 = X_0\ a$

$X_2 = X_1\ b$

$X_3 = X_2\ a$

$X_4 = X_0\ b$

$X_5 = X_1\ a\cdots$

と分解できる。増分分解した各成分系列は既成分系列を用いて次のような組で符号化する。

【0015】

【表2】

成分のインデックス（各成分の出た順番）	次のシンボル

【0016】すなわち、増分分解型アルゴリズムは、符号化パターンについて、過去に分解した部分列の内、最大長一致するものを求め、過去に分解した部分列の複製として符号化するものである。動的辞書型アルゴリズムの改良としては、

①LZW（Lempel-Ziv-Welch）符号（T. A. Welch, "A Technique for High-Performance Data Compression", Computer, June 1984 参照）

②LZJ符号（M. Jakobsson, "Comperssion of Character Strings by An Adaptive Dictionar, BIT, 25号，1985年，593-603頁参照のこと）

とがある。次にLZW符号について説明する。

［LZW符号］LZW符号の符号化の処理のフローを図14に示す。即ちLZW符号化は、書き替え可能な辞書をもち、入力文字コードのデータ中を相異なる文字列に分け、この文字列を出現した順に番号を付けて辞書に登録すると共に、現在入力している文字列を辞書に登録してある最長一致文字列の番号だけで表して、符号化するものである。尚、動的辞書型符号およびLZW符号の技術は、特開昭59-231683，米国特許4，558，302で開示されている。図14の符号化処理は次のようになる。

【0017】S1：予め全文字につき一文字からなる文字列を初期値として登録してから符号化を始める。辞書の登録数nを文字種数Aと置く。カーソルをデータの先頭の位置に置く。

S2：カーソルの位置からの文字列に一致する辞書登録の最長文字列Sを見つける。

【0018】S3：文字列Sの辞書番号を〔$\log_2 n$〕ビットで表して出力する。ただし、〔$\log_2 n$〕は$\log_2 n$以上の最小の整数である。辞書登録数nを一つインクリメントする。

S4：文字列Sにカーソルの最初の文字Cを付加した文字列SCを辞書に登録する。カーソルは文字列Sの後の文字に移動させる。S2に戻る。

【0019】図15はLZW符号の復号化を示したフローチャートであり、符号化の逆の処理となる。動的辞書型アルゴリズムは、辞書内の系列は過去に符号化した（サンプリングした）系列の中だけから選ぶため、処理速度が速い。しかし、過去に現れたデータの一部の系列しか含めないため圧縮率が高く取れない欠点がある。

【0020】動的辞書型アルゴリズムの改良版として、辞書への学習量を増やしインデックスのみで符号化できるようにしたLZJ符号がある。

（LZJ符号）LZJ符号の符号化の処理フローを図16に示し、また復号化の処理フローを図17に示す。

【0021】ここで、辞書と文字列の表記法を次のように定義する。文字種の集合をAとし、集合Aの文字を組み合わせてできる文字列をSで表す。文字列Sのi番目の文字をS（i）する。更に複数の部分文字列S（i），S（i+1），・・・，S（j）をS（i，j）とする。辞書をD_h（S）で表わし、辞書の木（tree）の根（root）から葉（leaf）へのパスとして文字列S中の一定の長さhの全ての部分文字列を登録する。

【0022】図16のLZJ符号化処理は次のようになる。

S1：辞書に全文字種の一文字を初期値として登録してから符号化を始める。辞書の登録数nを文字種数Aとおく。カーソルk＝0とおく。

S2～S5：k番目の入力文字まで符号化が終了したとして文字列S（1，k）の全ての部分文字列がすでに辞書D_h（S（1，k））に登録してある。S（k+1），・・・の文字列から符号化する。

【0023】詳細に説明すると、次のようになる。

S2：S（k+1），・・から辞書D_h（S（1，k））の登録文字列に最長一致する部分文字列S（k+1，k+z）を見つける。

S3：部分文字列S（k+1，K+z）の辞書番号a_Xを〔$\log_2 n$〕ビットで表して出力する。ただし、nは辞書の現在の登録数であり、〔$\log_2 n$〕は$\log_2 n$以上の最小の整数である。ここで、符号語a_Xは部

分文字列S（i_x，j_x）を表す。各々のa_xは辞書D$_h$（S（1，j_{x-1}））、（$i_x \leq j_x \leq i_x + h$，$i_x = j_{x-1} + 1$）の辞書番号である。

【0024】S4：部分文字列S（k−h+2，k+1），・・・，S（k+z−h+1，k+z）にnをインクリメントしながら辞書番号を付けて辞書に追加し、辞書D$_h$（S（1，k+z））を構成する。

S5：カーソルk＝k+zとおく。

S6：全文字を処理するまでS1～S5を繰り返す。

【0025】ここでステップS4の文字列の辞書登録を図示すると図18に示すようになる。次に図17のLZJ復号化処理は次のようになる。

S1：図16のS1と同様に辞書に全文字種の一文字を初期値として登録する。辞書の登録数nを文字種数Aとおく。カーソルk＝0とおく。

【0026】S2～S4：辞書番号a_wが復号化され、文字列S（1，j_w）まで利用することができ、辞書D$_h$（S（1，j_w））が再構成されている。次に符号語a_{w+1}を復号する。詳細に説明すると次のようになる。

S2：符号語a_{w+1}を復号した辞書番号より辞書D$_h$（S（1，j_w））内の部分列S（i_{w+1}，j_{w+1}）を復元する。部分列S（i_{w+1}，j_{w+1}）は辞書内で根（root）からアドレスa_{w+1}の節点で表わされる文字列である。

【0027】S3：文字列S（1，j_{w+1}）を復号した後、辞書D$_h$（S（1，j_{w+1}））を図16のS4と同様に構成する。

S4：カーソルk＝j_{w+1}とおく。

S5：全符号を処理するまでS1～S4を繰り返す。

【0028】

【発明が解決しようとする課題】しかしながら、従来のスライド辞書型アルゴリズムのLZSS符号および動的辞書型アルゴリズムのLZW符号は完全なユニバーサル性を前提にしており、辞書が空白の状態から符号化を始めるようにしている。このため、従来の符号化方式では、入力データの始めの方で、学習量が少ない（辞書内容が少ない）とき、圧縮率が低いという欠点があった。

【0029】LZW符号ではユニバーサル性も重要であるが、入力データに特定の種類のデータだけ特に多く現れるときは、辞書は必ずしも空白の状態から符号化する必要はない。この観点から本願発明者等は、動的辞書型アルゴリズムにおいて図19に示すように、高い頻度で出現する文字列のみ保持した辞書を用いて高圧縮率を得る方法を提案している。

【0030】図19にあっては、サンプルデータを対象にLZW符号化を行って辞書を作成し、辞書には使用頻度を同時に計数しておく。サンプルデータの符号化が済んだ場合には、辞書の中から出現頻度が閾値T以上の文字列を抽出して実際の符号化に使用する辞書に初期値として登録してから符号化又は復号化を行う。しかし動的辞書型アルゴリズムは符号化、復号化の処理速度はバランスしている。一方、スライド辞書型アルゴリズムにあっては、符号化は遅いものの復号化の処理が格段に速いため、データベースなどのように復元の処理が主になる用途においては有利となる。

【0031】本発明は、このような状況に鑑みてなされたもので、動的辞書型アルゴリズムを利用した初期値の登録でスライド辞書型アルゴリズムによる符号化を効率よくできるようにしたデータ圧縮方式を提供することを目的とする。

【0032】

【課題を解決するための手段】図1は本発明の原理説明図である。まず本発明は、入力バッファ（Qバッファ）10中の入力データを辞書バッファ（Pバッファ）12中の符号化済データの部分列の内、最長一致するものの格納位置と一致長で指定して符号化し、符号化済みの入力データは辞書バッファ12に移して新たな符号化済みデータとして次の入力データを符号化するスライド辞書型アルゴリズムに従ってデータ圧縮方式を対象とする。

【0033】このようなデータ圧縮方式につき本発明にあっては、代表的なサンプルデータを相異なる部分列に分けたとき、所定閾値T以上の出現頻度をもつ部分列を抽出し、この抽出した部分列を一列に並べて初期値文字列を予め作成する初期値作成手段14と、初期値作成手段14で作成した初期値文字列を、符号化又は復号化に先立って最初に辞書バッファ12に固定的に設定して符号化済又は復号化済データと見做し、初期値文字列および新たに設定した符号化済又は復号化済データの中から入力バッファ10の入力データと最長一致する部分列を検索して格納位置と一致長で指定して符号化又は復号化する符号化復号化手段16とを備えたことを特徴とする。

【0034】また本発明は、初期値作成手段14で作成した初期値文字列を、符号化に先立って最初に辞書バッファ12に固定的に設定して符号化済又は復号化済データと見做し、初期値文字列の中からのみ入力バッファ10の入力データと最長一致する部分列を検索して格納位置と一致長で指定して符号化し、辞書バッファ12に対する新たな入力データの登録は行わないようにしてもよい。

【0035】ここで、初期値作成手段14は動的辞書型アルゴリズムであるLZW符号の符号化処理に従って初期値文字列を作成する。即ち、符号化済み文字列を参照番号を付して登録する辞書を有し、代表的なサンプルデータの文字列に最長一致する辞書中の符号化済み部分列を検索して参照番号で指定して符号化し、この符号化後に参照番号に次のサンプル文字を付加した部分列を新たな参照番号を付して辞書に登録し、更に辞書に登録された符号化済み文字列の検索する毎に使用頻度を計数し、前記サンプルデータの符号化終了した際に前記使用頻度が所定閾値以上となる文字列を抽出し、抽出した文字列を出現順に並べて初期値文字列を予め作成する。

【0036】また初期値作成手段14は動的辞書型アルゴリズムであるLZJ符号の符号化処理に従って初期値文字列を作成してもよい。即ち、初期値作成手段14は符号化済み文字列を参照番号を付して登録する辞書を有し、代表的なサンプルデータの文字列に最長一致する辞書中の符号化済み部分列を検索して参照番号で指定して符号化する。この符号化後に符号化した入力文字列の各文字を順次接頭部分列とし、この接頭部分列に辞書中の部分列を加えて一定長の部分列を複数作成して全て辞書に登録し、更に辞書に登録された符号化済み部分列の検索する毎に使用頻度を計数し、サンプルデータの符号化終了した際に使用頻度が所定閾値以上となる部分列を抽出し、この抽出した部分列を出現順に並べて初期値文字列を予め作成する。

【0037】更に初期値作成手段（14）は、サンプルデータをスライド辞書型アルゴリズムで符号化して符号列を求め、この符号列に動的辞書型アルゴリズムであるLAW符号化やLZJ符号化を適用して使用頻度の高い初期値文字列を生成してもよい。即ち、入力バッファ中の代表的なサンプルデータの文字列を辞書バッファ中の符号化済データの部分列の内、最長一致するものの格納位置と一致長で指定して符号化し、この符号化データを相異なる部分列に分けたとき、所定閾値以上の出現頻度をもつ部分列を抽出し、抽出した部分列を出現順に並べて初期値文字列を予め作成する。

【0038】学習により作成した使用頻度の高い初期値文字列の辞書バッファ12への登録は、辞書バッファ12を読出専用の第1メモリ（ROM）と、読出及び書込可能な第2メモリ（RAM）で構成し、初期値作成手段14で作成した初期値文字列を第1メモリ（ROM）に固定的に記憶し、入力バッファ10の符号化済み入力データは第2メモリ（RAM）に移して記憶する。

【0039】また辞書バッファ12を書替え禁止領域を設定した読出及び書込可能なメモリ（RAM）で構成し、符号化開始時に初期値作成手段14で作成した初期値文字列を辞書バッファ12の書替え禁止領域にロードした後に符号化を開始するようにしてもよい。

【0040】

【作用】このような構成を備えた本発明のデータ圧縮方式によれば、次の作用が得られる。まず圧縮符号化しようとするデータの種類に応じたサンプルデータを対象に動的辞書型アルゴリズム、即ちLZW符号やLZJ符号のアルゴリズムに従った符号化を行うと共に、この符号化に使用する辞書にカウンタを設け、参照番号が符号化に使われた回数を使用頻度として計数する。

【0041】サンプルデータの符号化が済んだならば、辞書の登録済み文字列の気構造の接点に設けた使用頻度を示すカウンタ計数値の小さい文字列は辞書から削除し、高頻度で出現する文字列のみを残した辞書を求め、この文字列から一列に並べて初期値文字列を生成する。学習により生成された初期値文字列を用いたスライド辞書型アルゴリズムによる符号化は、次のようにする。

【0042】①符号化時に、予め記憶装置に取り出しておいた高頻度の初期値文字列を初期値として辞書にロードした後、符号化する。

②予め作成した高頻度の文字列を初期値として辞書の先頭に書き替えをしない固定部分として設定しておき、符号化する。スライド辞書型アルゴリズム、例えばLZSS符号による符号化であっても、使用頻度の高い文字列が予め辞書バッファにと初期値として登録されているため、最初から長い入力データの文字列に最長一致する辞書の文字列を検索することができ、スライド辞書型の符号化をより高速で行うことができる。

【0043】

【実施例】図2は本発明の一実施例を示した実施例構成図である。図2において、14は初期値作成部であり、代表的なサンプルデータを対象に動的辞書型アルゴリズムに従った符号化を行い、この符号化で作成される辞書に登録された文字列（部分列）を示す参照番号の符号化時に使われた回数を出現頻度としてカウンタで計数し、サンプルデータの符号化が終了した時点で所定閾値T以上の出現頻度をもつ辞書24の登録文字列（部分列）を取り出し、一列に並べて初期化文字列を作成する。

【0044】具体的には、初期値作成部14はサンプルデータ記憶部20，動的辞書型符号化部22，辞書24及び初期値文字列生成部26で構成される。サンプルデータ記憶部20にはデータ圧縮の対象となるデータの種類に応じた代表的なサンプルデータが学習対象として記憶される。動的辞書型符号化部22は動的辞書型アルゴリズムに従ってサンプルデータ記憶部20のサンプルデータを対象に辞書24を作成しながら符号化を行う。

【0045】この動的辞書型アルゴリズムとしては、例えば図13に示したLZW符号化アルゴリズム、あるいは図15に示したLZJ符号化アルゴリズムを用いることができる。辞書24は参照番号に対応して符号化済み文字列としての部分列を登録しており、更に符号化済み部分列の参照番号が符号化時に使われた回数を計数するカウンタを設け、出現頻度として計数するようにしている。初期値文字列生成部26は動的辞書型符号化部22によるサンプルデータの符号化処理が終了した段階で辞書24の中から所定の閾値T、例えばT＝2以上となる出現頻度をもつ部分列を取り出し、この部分列を一列に並べて使用頻度の高い初期値文字列を生成する。

【0046】初期値作成部14で作成された初期値文字列はデータ圧縮復元部30におけるスライド辞書を用いたデータ圧縮に用いられる。データ圧縮復元部30は入力バッファとしてのQバッファ10、辞書バッファとしてのPバッファ12及びスライド辞書型符号化復号化部

１６で構成される。Ｐバッファ１２には符号化及び復号化に先立って初期値作成部１４で予め作成された初期値文字列が登録され、この初期値の登録領域はＱバッファ１０から入力文字列をシフトして格納しても廃棄されず、固定的に保持される。即ち、Ｐバッファ１２に登録した初期値文字列を符号化済みデータと見做してスライド辞書型アルゴリズムに従った符号化及び復号化を行う。

【００４７】スライド辞書型符号化復号化部１６はスライド辞書型アルゴリズムに従った符号化または復号化を行うもので、具体的には、ジブーレンペル符号化アルゴリズムや図１０に示したＬＺＳＳ符号化アルゴリズムを実行する。図３は図２の初期値作成部１４の処理を示したフローチャートである。図３における初期値作成処理は次のようになる。

【００４８】Ｓ１：入力データとして多く出現するデータサンプルを入力し、ＬＺＷ符号またはＬＺＪ符号等の動的辞書型アルゴリズムに従ってサンプルデータの符号化を行う。この符号化において、木構造の辞書が作成される。同時に符号化により作成される辞書の木構造における各節点が文字列を表すことになるが、各節点にカウンタを付加して符号化時に最長一致する文字列を検索したときに通過した各節点でカウンタを１つインクリメントして使用回数を計数する。

【００４９】即ち、最長一致の文字列を検索した場合には、検索した文字列に含まれる節点のカウンタの全てがカウントアップされることになる。

Ｓ２：サンプルデータの符号化が終了した時点で辞書の各節点の連鎖で構成される文字列の中から節点に設けたカウンタの計数値が所定の閾値Ｔ以上の高頻度で使用された文字列を取り出す。

【００５０】Ｓ３：Ｓ２で取り出した文字列を一列の文字列の形に並べ、初期値文字列を生成する。このとき既に並べた文字列の中に新たに取り出した文字列と同じ文字列があるか否かを検索し、もし同じ文字列があれば冗長するので、初期値の文字列には含めないようにする。以上のＳ１～Ｓ３の処理を経て作成された初期値文字列はスライド辞書型アルゴリズムに従った符号化及び復号化に使用するため、外部の補助記憶装置等に取り出しておくことが望ましい。

【００５１】図４は図３の初期化処理のステップＳ１における動的辞書型アルゴリズムに従ったサンプルデータの符号化の際に作成される辞書の木構造とその節点に設けたカウンタによる使用頻度の計数を示した説明図である。図４にあっては、まず辞書に例えばａｂｃｄの各文字を参照番号①～④に示すように初期登録した後にサンプルデータの符号化を開始しており、図示の状態にあっては更に参照番号⑤～⑨までの文字列の登録が済んだ状態での木構造を示している。

【００５２】例えば、入力データａｂｃの符号化は参照番号①～⑦で示す文字列の登録が済んだ段階で行われており、文字列ａｂは辞書２４の検索により参照番号⑥で示す文字列に一致していることから、出力符合は「⑥ｃ」として出力し、続いて参照番号⑥に次の１文字ｃを加えた文字列を新たに参照番号⑧を付して辞書２４に登録している。

【００５３】次の文字列ａｂｄについても、辞書２４の検索で参照番号⑥の文字列に最長一致し、従って出力符合は「⑥ｄ」として出力され、辞書２４に対しては参照番号⑥に次の１文字ｄを加えた文字列に新たな参照番号⑨を加えて登録している。この文字列ａｂｃと文字列ａｂｄの符号化後の登録にあっては、同じ文字列ａｂが２回使用されているため、参照番号①の節点及び参照番号⑥の節点の各カウンタが２回カウントアップされ、それぞれ５，３となる。

【００５４】このような木構造を備えた辞書２４における各節点の計数値は、その節点の子供の計数値の和に１を加えた値となる。例えば、参照番号①の文字ａの節点の計数値はその節点の子供となる参照番号⑤と⑥の計数値の和１＋３＝４に１を加えた値として５となる。図５は図３のステップＳ２及びＳ３に示したサンプルデータの符号化で得られた辞書から閾値Ｔ以上の高頻度で使用された文字列を取り出して、初期値として使用する文字列の形に変換する処理を示した説明図である。

【００５５】図５（ａ）はサンプルデータの符号化が終了した状態で得られた辞書の木構造を示したもので、各文字の節点に設けたカウンタ内の計数値が使用頻度を示している。この図５（ａ）の辞書の木構造に対し、閾値Ｔ＝２以上の計数値をもつ文字列を取り出すと、図５（ｂ）に示すようになる。この閾値２以上の計数値をもつ文字列は、図５（ｃ）に示すように、例えば左側の文字列から順番に１本の文字列にまとめるように並べ替えられ、スライド辞書型アルゴリズムのＰバッファに初期値として登録される初期値文字列が作られる。

【００５６】図６は図２のデータ圧縮復元部３０におけるスライド辞書を用いた符号化処理を示したフローチャートであり、次のようになる。

Ｓ１：図３の初期値作成処理で作られた初期値文字列（Ｎ文字）をＰバッファの前半に格納する。

Ｓ２：通常のスライド辞書型アルゴリズムと同様に、入力したＱバッファ１０の文字列を初期値文字列と符号化済み文字列を格納したＰバッファ１２から最長一致する文字列を検索して、開始位置と一致長の組で符号化する。この場合、符号化文字列は２文字一致するか否かによって次の２つのモードで符号化される。

【００５７】符号化モード［識別ビット０］［最長一致文字列の位置］［一致長］

生データモード［識別ビット１］［１文字］

Ｓ３：辞書の削除及び登録処理として初期値文字列を除くＰバッファ１０内の部分をスライドさせる。即ち、Ｐバッファ１０内の初期値文字列を示す０～ｎ－１の位置の文字はそのままにしておき、文字位置ｎから符号化が済んだＱバッファ１０の文字数だけ文字を左にシフトして削除し、新たにＱバッファの符号化済み文字列をＰバッファ１２の左側から右側にシフトして追加する。

【００５８】以上の辞書削除及び登録処理が済んだならばＱバッファ１０に符号化が済んだ文字分を左にシフトし、新たな文字列を入力する。以下同様に、ステップＳ２，Ｓ３の処理を繰り返して入力文字列を符号化する。図７は図６のスライド辞書型アルゴリズムを用いた符号化におけるＰバッファの構成を示した説明図である。

【００５９】図７（ａ）はＰバッファ１２として書替え可能なメモリ、例えばＲＡＭを使用した場合であり、Ｐバッファ１２の前半の斜線部で示す０～ｎ－１の領域に初期値文字列ロード領域を設け、ここに予め作成した初期値文字列を格納し、残りを符号化済み文字列の登録領域としている。この初期値文字列ロード領域０～ｎ－１についてはその後の書替えを禁止し、新たなＰバッファ１２に対する登録に際しては、ｎの位置から、右から左に登録文字数分だけのシフトに伴って、登録していた文字を処理済み文字列として排出して削除する。

【００６０】図７（ｂ）はＰバッファ１２の他の構成を示したもので、この実施例にあっては、初期値文字列を登録する領域として固定記憶を実現するため、ＲＯＭを用いたＲＯＭ領域としており、残りの領域を書替え可能なＲＡＭを用いたＲＡＭ領域を設けている。Ｐバッファ１２のＲＡＭ領域はＱバッファ１０における入力文字列の符号化が１つ加わる毎に符号化済み文字数分の文字がＲＡＭ領域の左端から捨てられ、Ｑバッファ１０の符号化済み文字列がＲＡＭ領域の右端よりシフトして格納される。

【００６１】図８は図２のデータ圧縮復元部３０による復号化処理を示したフローチャートであり、次のようにして行われる。

Ｓ１：図６のステップＳ１と同様、Ｐバッファ１２の前半に予め作成された初期値文字列（ｎ文字）を格納する。

Ｓ２：符号語を入力し、複製モードのときはＰバッファ１２を参照して文字列を復元する。

【００６２】Ｓ３：図６のステップＳ３と同様に、Ｐバッファ１２の削除と登録の処理を行う。以下同様に、ステップＳ２，Ｓ３の処理を繰り返して、符号化した文字列を復元する。図９は本発明の初期値文字列の作成処理の他の実施例を示した説明図である。図９の初期値文字列の作成処理にあっては、まずスライド辞書型アルゴリズムに従ってサンプルデータを符号化し、サンプルデータから得られた符号列を対象に動的辞書型アルゴリズムに従った符号化を行い、この符号化の際に作成される辞書の木構造における節点にカウンタを設けて使用頻度を計数し、符号化が終了したならば所定の閾値Ｔ以上の使用頻度をもつ文字列を取り出し、１列に並べて初期値文字列を作成する。

【００６３】即ち、まずサンプルデータをＱバッファ１０ａに入力し、Ｐバッファ１２ｂの登録済み文字列を検索して最長一致する文字列を求め、開始位置ｐｉと一致長ｑｉで符号化する。勿論、この符号化は２文字以上の場合に行われ、１文字の場合は生データを出力する。このようなスライド辞書型アルゴリズムによる符号化で出力符号列Ｓ１，Ｓ２，・・・Ｓｉ，・・・が得られたならば、この符号列Ｓ１，Ｓ２，・・・Ｓｉ，・・・を対象に動的辞書型アルゴリズムに従った木構造の辞書２４を作成する。

【００６４】辞書２４にあっては、符号列Ｓｉの設定毎にカウンタが設けられ、文字列の符号化に使用された回数が計数されている。符号列Ｓ１，Ｓ２，・・・のＬＺＷ符号化が済んだならば、辞書２４の中の例えば閾値Ｔ＝２以上の符号列を取り出し、１列の符号列に並べ替えて初期値符号列として例えばＳ１，Ｓ２，Ｓ３，・・・を生成する。ここで、初期値符号列Ｓ１，Ｓ２，Ｓ３，・・・の各文字列は最初の符号化の際に予め判っていることから、元の文字列に復元することで初期値文字列を生成する。

【００６５】このように最初にスライド辞書型アルゴリズムを用いてサンプルデータから符号列を作り、この符号列を対象に初期化作成を行うようにした場合には、ＬＺＷ符号化アルゴリズムのみで辞書を作成した場合に比べ、辞書に登録される文字列の最大長に制約があるかどうかの違いだけであり、基本的にはＬＺＪ符号化で作成した辞書に基づく初期値文字列の作成と略同じものが得られる。

【００６６】更に本発明の他の実施例としては、データ圧縮を行おうとする入力データの種類が予め判っているときには、スライド型辞書としてのＰバッファ１２を初期値作成部１４で作成された初期値文字列のみとしてもよい。このように、Ｐバッファ１２の登録を初期値文字列のみとした場合には、圧縮率はやや劣るものの復号化の処理が簡単になり、従来のスライド辞書型アルゴリズムにおいて復号化に時間がかかるという問題を解決し、従来方式に比べ非常に高速な処理を実現することができる。

【００６７】更に本発明の他の実施例として、出現するデータの種類が予めだいたい予想できるような場合には、初期値文字列をデータの種類によって何種類か準備しておき、初期値文字列の部分を取り替えることによって、予想した特定種類のデータに強い圧縮効果が得られる方式を実現することができる。

【００６８】

【発明の効果】以上説明してきたように本発明によれば、出現頻度の高い特性種類のデータでなる初期値文字列を辞書に登録することによって、ＬＺＳＳ符号毎のス

ライド辞書型アルゴリズムの簡便さをほとんど変えることなしに圧縮率を高めることができる。

【0069】また、初期値文字列にない出現頻度の少ない種類のデータについては、初期値文字列を設定した後の辞書空きスペースに符号化進む毎に新たな文字列を登録することによって出現頻度の低いデータについてもユニバーサル性を損うことなく圧縮できる。

【図面の簡単な説明】

【図1】本発明の原理説明図

【図2】本発明の実施例構成図

【図3】本発明の初期値作成処理を示したフローチャート

【図4】本発明のサンプルデータを対象とした動的辞書型符号化と辞書の文字列使用回数の計数を示した説明図

【図5】本発明の符号化辞書の使用頻度に基づく初期値文字列の生成を示した説明図

【図6】本発明の初期値文字列を用いたスライド辞書型符号化のフローチャート

【図7】本発明のスライド辞書型符号化で使用するPバッファの構成を示した説明図

【図8】本発明のスライド辞書型復号化のフローチャート

【図9】サンプルデータをスライド辞書型符号化で符号列に変換した後に木構造の動的辞書を作成して初期値文字列を生成する本発明の他の実施例を示した説明図

【図10】スライド辞書型符号化の原理図

【図11】従来のＬＺＳＳ符号化アルゴリズムを示したフローチャート

【図12】ＬＺＳＳ符号化に用いるバッファ構成図

【図13】ＬＺＳＳ符号化の符号化データの出力形式説明図

【図14】従来のＬＺＷ符号化アルゴリズムを示したフローチャート

【図15】従来のＬＺＷ復号化アルゴリズムを示したフローチャート

【図16】従来のＬＺＪ符号化アルゴリズムを示したフローチャート

【図17】従来のＬＺＪ復号化アルゴリズムを示したフローチャート

【図18】ＬＺＪ符号化における文字列の登録を示した説明図

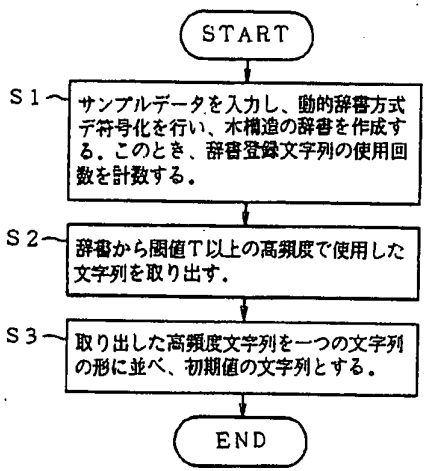
【図19】本願発明者が既に提案しているＬＺＷ符号を用いたデータ圧縮における辞書の初期登録の説明図

【符号の説明】

１０：入力バッファ（Ｑバッファ）
１２：辞書バッファ（Ｐバッファ）
１４：初期値作成手段（初期値作成部）
１６：符号化復号化手段（スライド辞書型符号化復号化部）
２０：サンプルデータ記憶部
２２：動的辞書型符号化部
２４：辞書
２６：初期値文字列生成部
３０：データ圧縮復元部

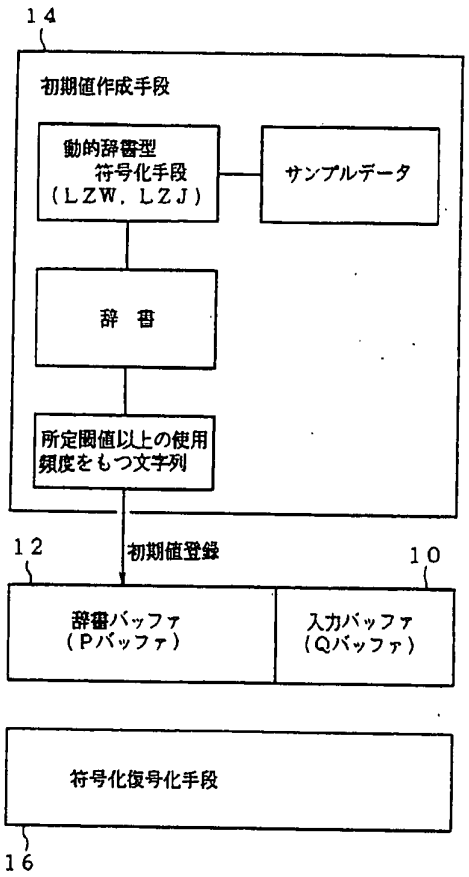
【図3】

本発明の初期値作成処理を示したフローチャート

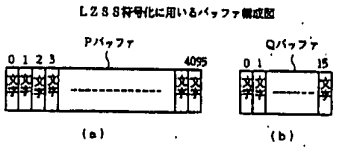


【図1】

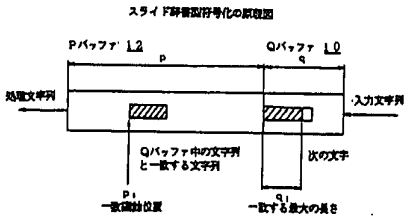
本発明の原理説明図



【図12】

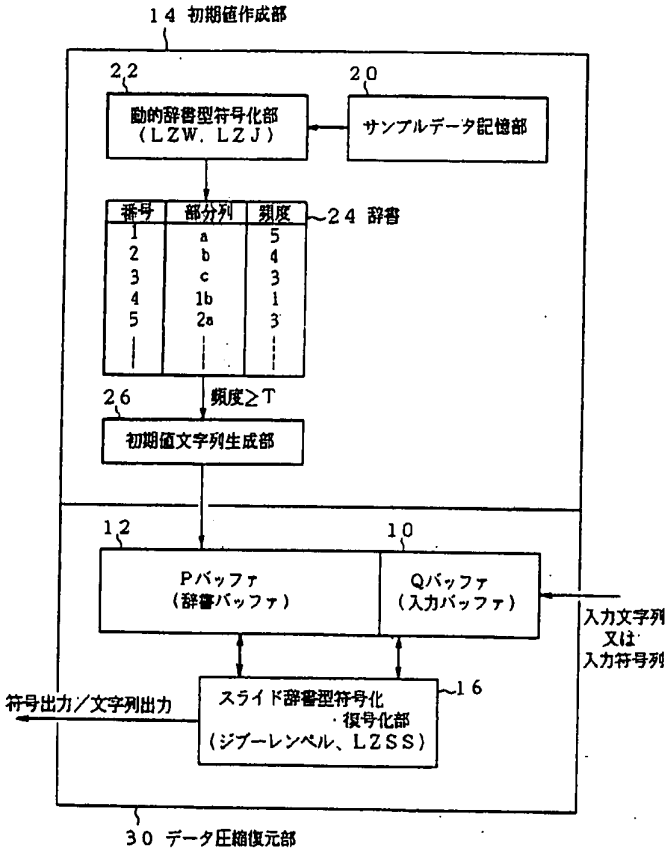


【図10】



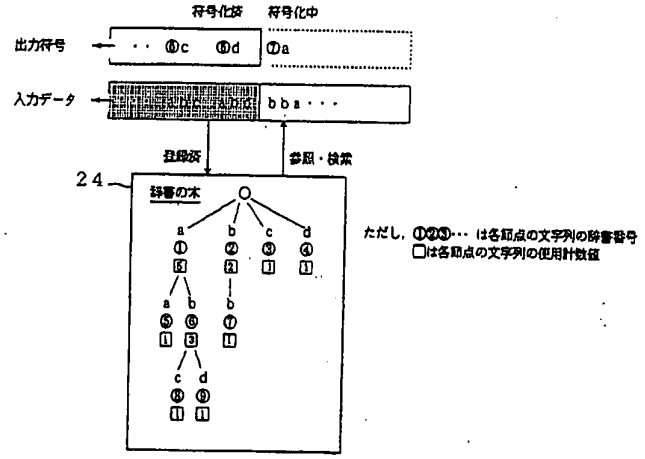
【図2】

本発明の実施例構成図



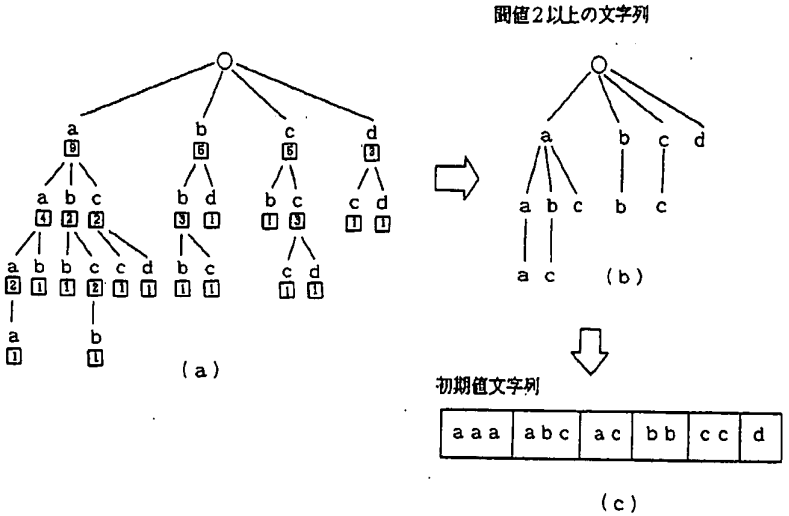
【図4】

本発明のサンプルデータを対象とした動的辞書型符号化と辞書の文字列使用回数の計数を示した説明図



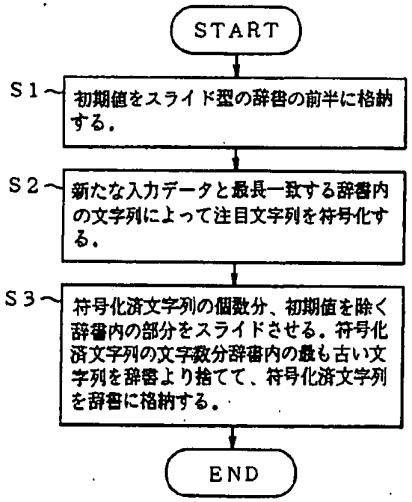
【図5】

本発明の符号化辞書の使用頻度に基づく初期値文字列の生成を示した説明図



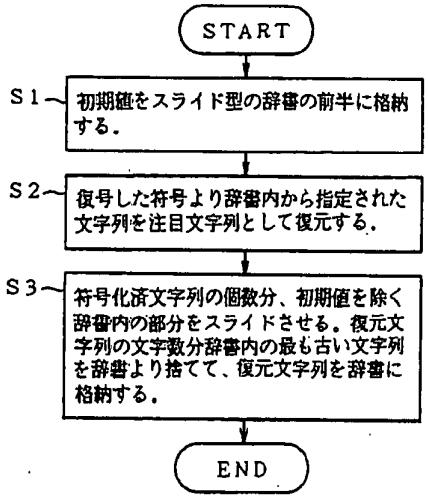
【図6】

本発明の初期値文字列を用いたスライド辞書型符号化のフローチャート



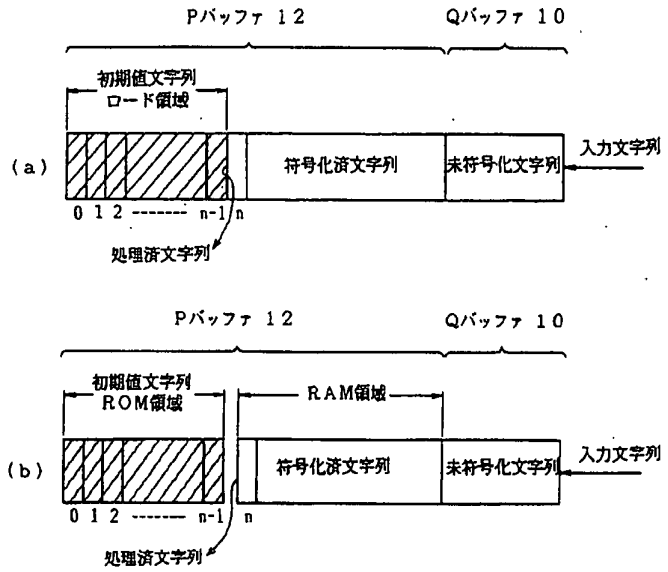
【図8】

本発明のスライド辞書型復号化のフローチャート



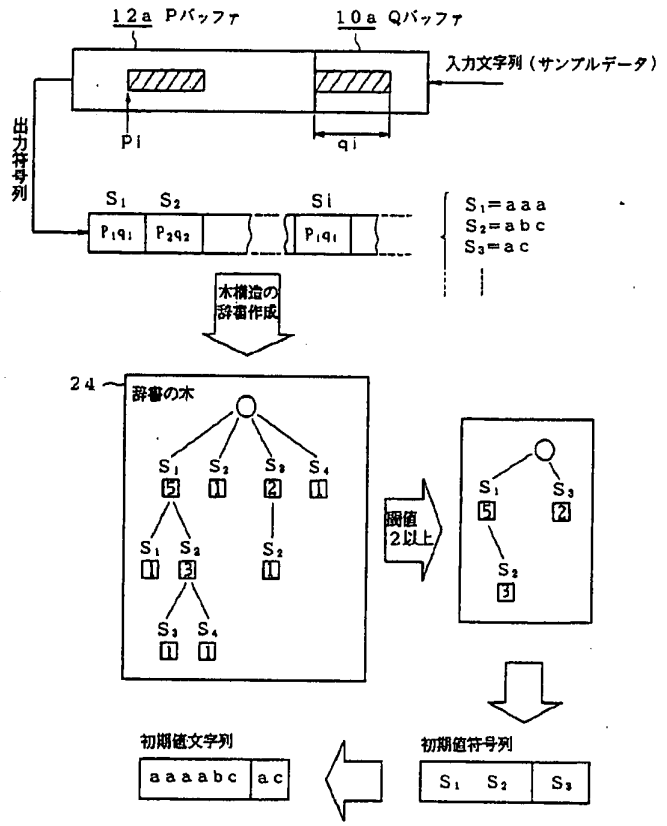
【図7】

本発明のスライド辞書型符号化で使用するPバッファの構成を示した説明図



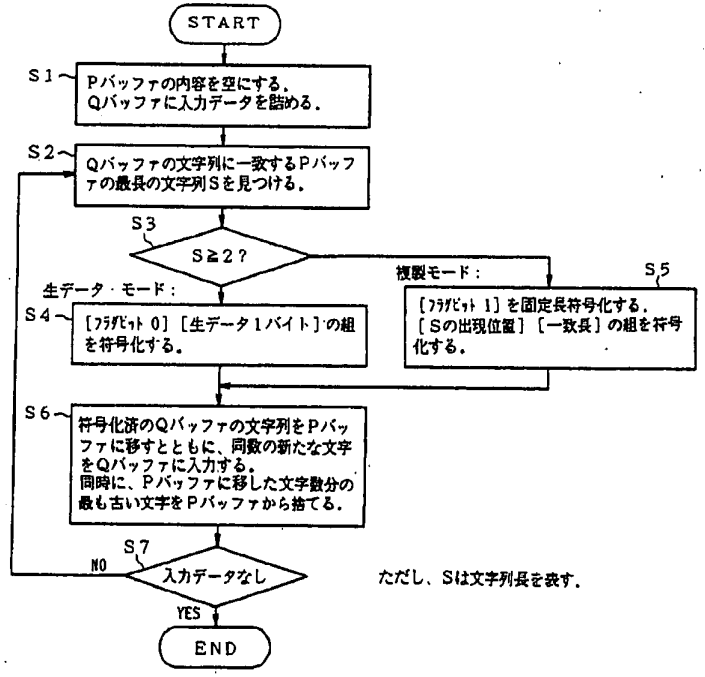
【図９】

サンプルデータをスライド辞書型符号化で符号列に変換した後に木構造の動的辞書を作成して初期値文字列を生成する本発明の他の実施例を示した説明図



【図１１】

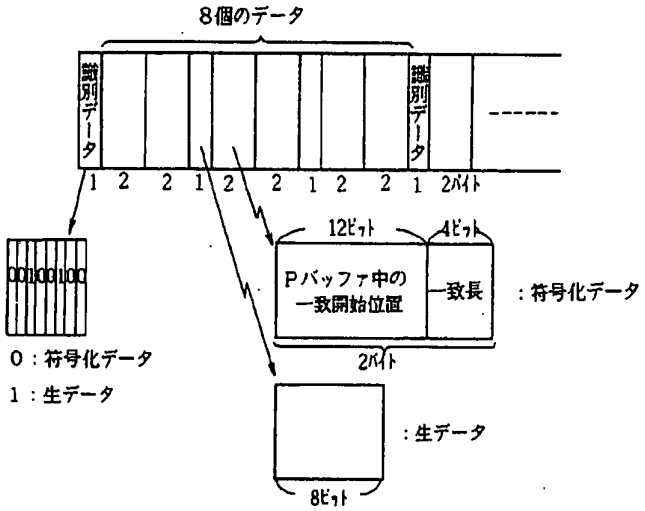
従来のＬＺＳＳ符号化アルゴリズムを示したフローチャート



ただし、Sは文字列長を表す．

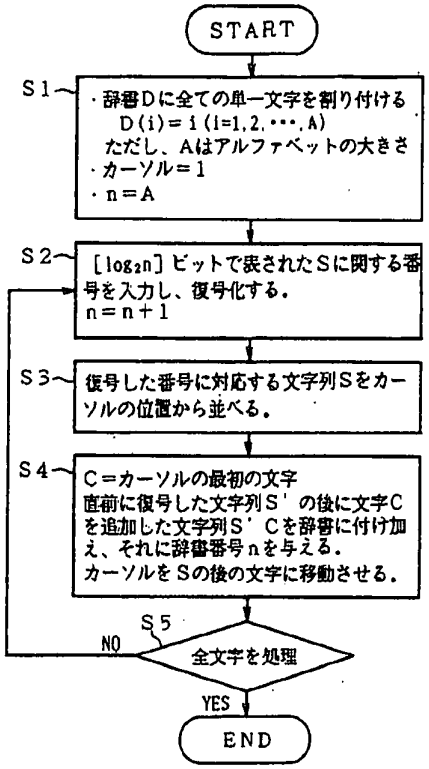
【図１３】

ＬＺＳＳ符号化の符号化データの出力形式説明図



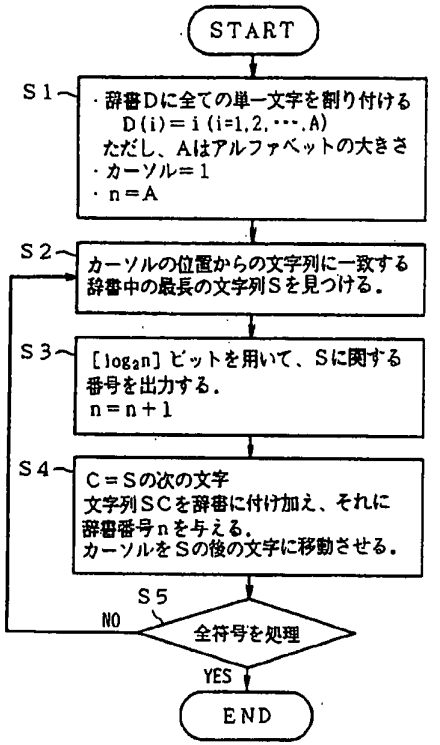
【図１４】

従来のＬＺＷ符号化アルゴリズムを示したフローチャート



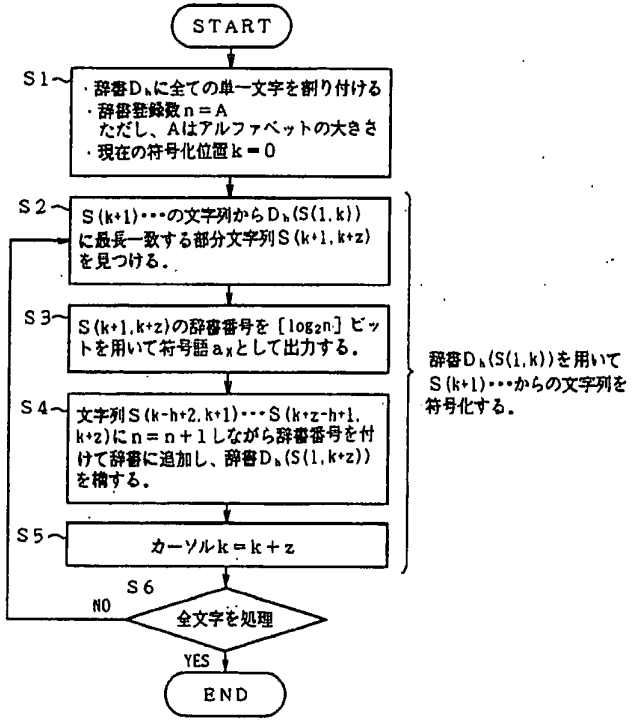
【図15】

従来のＬＺＷ復号化アルゴリズムを示したフローチャート



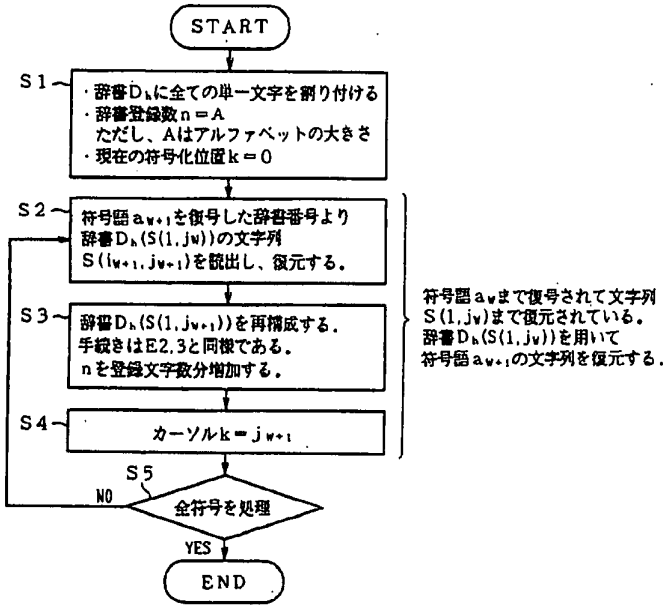
【図16】

従来のＬＺＪ符号化アルゴリズムを示したフローチャート

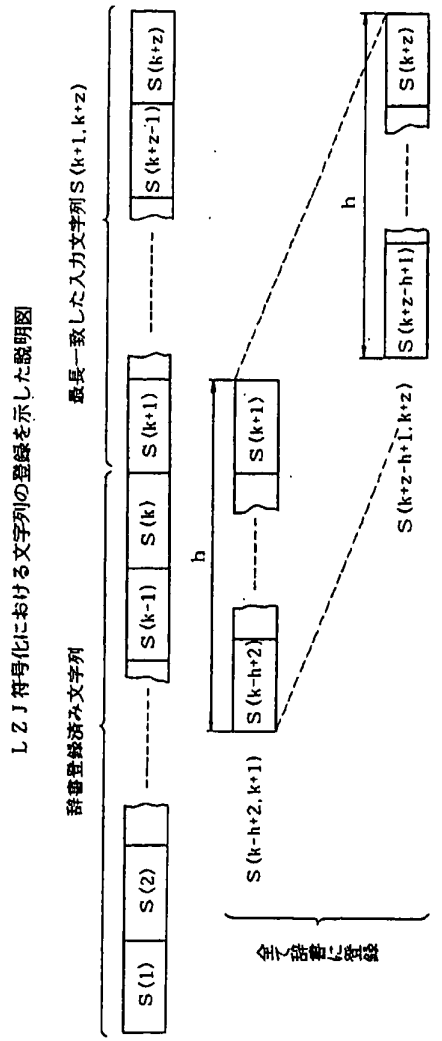


【図１７】

従来のＬＺＪ復号化アルゴリズムを示したフローチャート

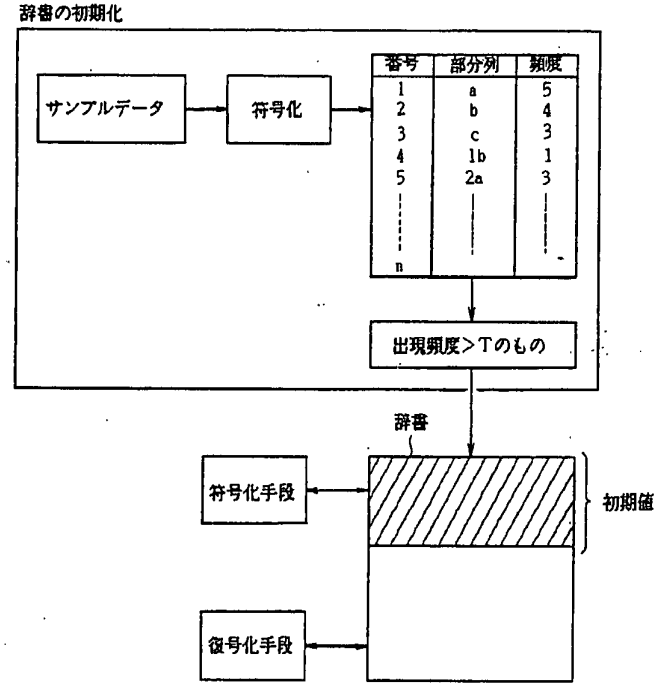


【図１８】



【図１９】

本願発明者が既に提案しているＬＺＷ符号を用いたデータ圧縮における辞書の初期登録の説明図



フロントページの続き

(72)発明者　千葉　広隆
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内